UDC: 004.8 https://doi.org/10.15407/jai2025.01.107

O. Lukianykhin[1], V. Shendryk[2]
[1,2]Sumy State University, Ukraine
116, Kharkivska St., Sumy, 40007
[1]oleh.lukianykhin.a@gmail.com
[2]v.shendryk@cs.sumdu.edu.ua
[1]https://orcid.org/0000-0002-4211-2401
[2]https://orcid.org/0000-0001-8325-3115

MACHINE LEARNING-DRIVEN PHOTOVOLTAIC GENERATION FORECASTING FOR PROSUMER DECISION SUPPORT

Abstract. The problem of forecasting electricity generation is key to enabling decision-making support at the level of individual prosumers in the power grid and efficient prosumer integration into the grid. This study investigates the application of machine learning (ML) approaches to photovoltaic generation forecasting, aiming to provide general practical recommendations for the development of applied forecasting solutions. To this end, a specific use case was considered in the context of a private household with a photovoltaic installation, where data was gathered for several years. Based on the experimental results, a set of recommendations for applying ML models to photovoltaic generation forecasting tasks was formulated in the context of prosumer decision support. These recommendations address key aspects such as training and test data sizes used in the model creation process, and prediction horizon size used in the prediction process. In addition, guidelines on model file size were developed from the perspective of practical model utilization in specific use cases. This research demonstrates that establishing universal guidelines for ML model utilization in the Power System (PS) domain is both beneficial and achievable. It also highlights opportunities for further research on developing solutions for automated recommendations for required training data sizes and prediction horizons.

Keywords: machine learning, photovoltaic generation, generation forecasting, decision support, prosumer.

Introduction

The modern state of the power grid and the most likely directions of the grid development are characterized by several structural changes in the power system (PS). One of them is the increasing number of renewable energy sources installed at a micro level of the grid, e.g., in private households [1–5]. Depending on the geography and climate, individual wind turbines and photovoltaic installations are one of the most popular [6,7].

This trend leads to the emergence of prosumers in the grid - consumers with some generation capacity [8–11]. However, because of the small scale and other factors, such as data safety concerns, decision-making at this grid level is complex [2,8–14]. Besides, forecasting is a key component of further decision-making, as effective planning and control are arguably impossible without available estimates of the generation and consumption volumes [12,15–20].

The object of the study is forecasting photovoltaic electricity generation for prosumers' decision-making support. Developing an applied forecasting solution, such as photovoltaic generation forecasting, is often time-consuming and requires significant efforts to achieve the required performance levels. In particular, the development process is slowed down because such problems are frequently considered from scratch, with limited reuse or knowledge transfer from similar use cases [18,19,21]. In addition, sufficient expertise in the PS domain is required to this end, especially when classic analytical approaches are applied. Therefore, it is essential to research efficient ways of developing applied forecasting solutions in the domain.

The subject of the study is the application of machine learning (ML) models to the aforementioned forecasting problem. The ML approach softens the requirements for domain knowledge. However, one should have a sufficient understanding of the best practices for ML applications instead. Otherwise, a modest or even inferior performance may be achieved compared to the classic analytical approaches [15,21,22]. Hence leading to the conclusion of the irrelevance of ML methods to the problem. This may limit the development of potent solutions for the forecasting problem and domain development in general.

The purpose of the work is to investigate the application of ML approaches to photovoltaic generation forecasting to provide practical recommendations for the development of such applied solutions. This problem is considered in a specific use case of the household with a photovoltaic installation and in the context of prosumer decision support.

Problem Statement

It is crucial to match electricity supply and demand inside the electrical system, but as both profiles are highly stochastic, there is a bit of uncertainty in this planning process. Decision-making in prosumers relies heavily on understanding the upcoming electricity generation power.

Therefore, forecasting photovoltaic electricity generation becomes a core problem of the corresponding decision support system (DSS) [12,15–20]. The problem can be summarized as follows: the goal is to predict the amount of energy generated at a specific time period in the future, given the information that is available to the DSS on the prosumer side. This goal has to be achieved despite the inherent stochastic nature of the processes and dependency on hardly predictable external factors.

Formally, given a set of historical data pairs (x; y), where x is the input features, and y is the target variable, the objective is to find an estimator F that effectively maps F(x) -> y, even for previously unseen data. As for details, including effectiveness criteria, see Materials and Methods Section.

Specific features used for forecasting may vary, as well as the amount of historical data available for model training or the time period in the future that one forecasts for. Moreover, these parameters can be changed by the developer of a solution on purpose.

In every case, such parameters of the considered problem may substantially impact the final solution and its performance. Thus, the research questions set in this study were:

1. What is the optimal amount of training and testing data for ML model application in this task?

2. What is the optimal prediction horizon?

3. What are the requirements of ML models regarding model size, and how does its limitation impact performance?

Related Work

Efficient decision support is crucial for prosumers' functioning and integration into a power grid [2,8–10,12–14]. Its core component is usually a set of forecasting solutions [12,15–20]. Prosumer’s decision-making needs insight into future system behavior, including the expected supply and demand [15,20,22–24]. That is, forecasting enables prosumers to manage demand-supply balance, constrained system resources, and financial goals with efficient decision-support systems [12,15,16,20,25,26].

Traditionally, forecasting problems were solved using classic analytical approaches. This also applies to renewable generation forecasting, as demonstrated in [21,27–29]. However, as seen in [27,28,30], such methods often require significant domain knowledge and effort to construct and maintain the solution. These limitations are particularly problematic for the highly variable nature of the processes in micro-grids and on the prosumer level [11,18,20].

In recent years, ML has proven to be a powerful tool for solving various forecasting problems, including those in the PS domain [15,19,21,22,27,30–33]. Some of its main benefits are the ability to process large amounts of historical data and find complex patterns in that data that traditional methods could overlook [28,33]. Arguably, classic analytical approaches are built on the assumption of specific patterns present in the data. Hence, a person developing such a solution should have deep domain expertise, while the algorithm is not capable of detecting new patterns by design [27,28,30].

Therefore, ML approaches not only show superior performance in many forecasting tasks, as shown in [19,22,27,30–32], but also reduce the requirements for domain knowledge [28,33]. Various ML-based solutions are successfully developed for renewable energy generation forecasting [21,30,33], including individual household levels [20,34].

However, applying the ML approaches

brought its own challenges to the domain. In particular, it is crucial to understand all the particularities and best practices of ML approach application. Otherwise, achieving the required performance level is hard, as demonstrated in [35]. This is a particularly acute problem for developing applications for a smaller scale of the grid, e.g., individual household size prosumers. Depending on the correctness of the approach, ML models may demonstrate different performance levels. At the same time, forecasting for decision support requires stable and consistent model behavior, as mentioned in [16]. Thus, applying ML approaches properly is mandatory by considering all the related aspects.

The researchers are considering certain applied aspects of the ML solution development. For instance, it is already evident that different ML algorithms have different requirements for the quality of the data and its availability [36]. At the same time, there are many such aspects, e.g., proper experiment and evaluation procedures, model type choice, data preparation, feature engineering techniques, model hyperparameters choice, etc. [36–38]. Some of those are considered partially and often in the context of a particular applied task while still being crucial for successful solution development, as in [39–41].

This leads to a situation when the PS domain is experiencing a surge of various ML-driven research projects [19,21,22,27,30–32]. Still, most are specific applied use cases, not trying to generalize. Therefore, finding recommendations or best practices for ML-based solution development in the PS domain is hard. However, one can find either general recommendations [42] or recommendations for other domains [40,41]. Nevertheless, such recommendations would be highly beneficial for developing applied solutions and further research. Because of that, the focus should be shifted, and more research is needed on how to apply ML methods in forecasting for prosumer decision support properly.

Materials and Methods

As this paper focuses on investigating efficient application of ML methods to generation forecasting, it is essential to describe the main components of an ML-powered solution, i.e., data, tools, and modelling approach.

A dataset was gathered in a private household with a photovoltaic generation unit, battery, and access to the main power grid. Data was collected at a daily frequency. It has a total of 1774 data points.

Exploratory data analysis was conducted to understand the data better. The target variable is the amount of generated energy in kWh’s. Other than the target variable, the considered dataset contains several other features available at the prosumer system controller level:

– date;
– outside temperature at 6:00;
– outside temperature at 18:00;
– consumption from the grid;
– amount of energy used from the battery.

As real-world data, this dataset has certain particularities. One of the most important ones is the presence of significant gaps, especially in the first half of 2019 and at the end of 2023. These data collection gaps are long and sequential, so correcting them effectively is hard [36].

Because of that, the data was filtered at the very beginning to include only the period of 1 Jan 2020 - 10 Nov 2023 for experiments.

At the same time, small gaps of missing values were still present in the data. As some of the employed algorithms required continuous data, e.g., time-series approaches, data imputation techniques were used. The following approaches were considered for imputing missing values in the target variable:

– filling empty values with zeros - the most straightforward possible approach used as a baseline for comparison;
– filling empty values with a median of the target variable in historical data - a method powered by domain knowledge, which is easy to interpret;
– polynomial interpolation using a polynomial of degree 3 - a more sophisticated method.

After considering several examples of gaps from the data (See Fig. 1), it became evident that median value is the optimal

option, as it is a sufficiently lightweight approach that is well suited based on domain knowledge for this problem. So, it was used later for all the experiments for the target variable. However, polynomial interpolation of degree 2 was used for other numerical features in the dataset.

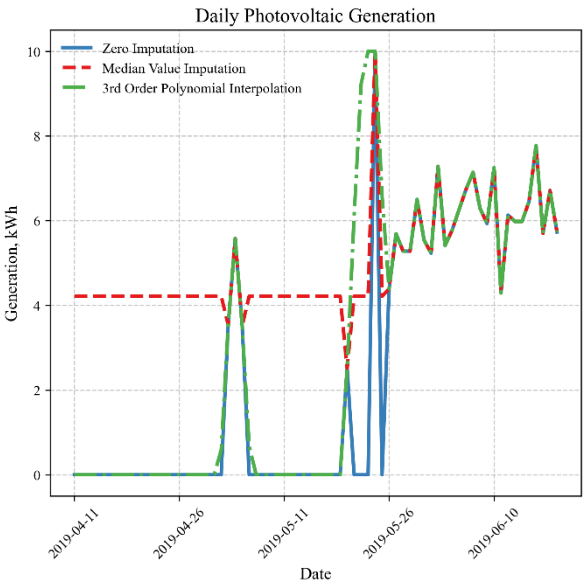


Figure 1. Comparison of empty values imputation results for the target variable (generation in kWh)

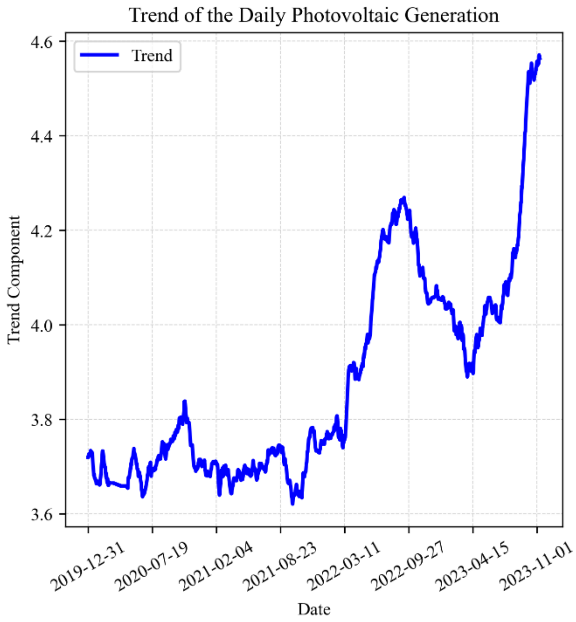
In addition, the final values of a target variable were cleaned from outliers based on the IQR approach and domain knowledge, e.g., the generation value can't be less than 0 in this setup. The resulting lower and upper bounds for cropping values were 0 and 10, respectively.

After these transformations, the main descriptive statistics of the considered target variable - amount of generated energy in kWh 's - are as follows: minimum value - 0.09, maximum value - 10, standard deviation - 2.5, and quantiles 0.25, 0.5, 0.75 correspondingly - 1.6, 4.21, 5.94.

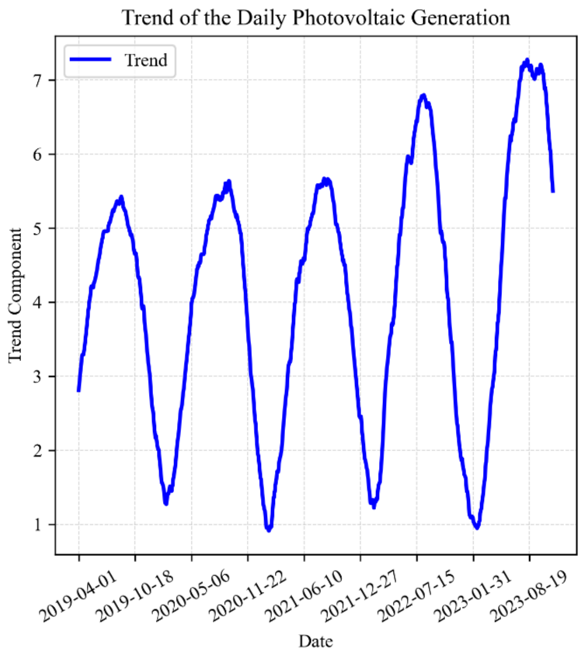
Most ML models can't consume dates directly, so a feature engineering step was conducted to generate the date-related features. In addition, some other features were created to enhance the model's capability to use the provided information for effective forecasting. The list of created data-related features:

– day of the week;
– is weekend;
– day of the month;
– month;
– temperature change.

Based on recent research [43] strong evidence of yearly seasonality is often observed in daily energy-related data, i.e., generation or consumption. A best practice from the time series analysis field was used to check this hypothesis - series decomposition into a trend, seasonal component, and noise was conducted.



(a)



b)

Figure 2. Trend component of the time series extracted in the decomposition process with the expected seasonality of a) 365 days and b) 90 days

The following results were received for

seasonality of 365 days and multiplicative decomposition (See Fig. 2a). In the case of other possible seasonality, decomposition results in a trend with a clear repetitive pattern, e.g., 90 days in Fig. 2b.

Because of that, two more date-related features were added. Their purpose was to allow ML models to learn yearly seasonality components. The formulas (1) and (2) were utilized:

$$day\_of\_year\_\sin = \sin\left(\frac{2\pi * day\_of\_year}{365}\right) \quad (1)$$

$$day\_of\_year\_\cos = \cos\left(\frac{2\pi * day\_of\_year}{365}\right) \quad (2)$$

To make sure the assumptions for the application of time series approaches are satisfied, the Augmented Dickey-Fuller (ADF) test was conducted for stationarity check. The ADF Statistic was -3.30, and the corresponding p-value was 0.0147, which indicated that the series is stationary as $p<0.05$. With that assumption checked, numerous time series models can be applied out of the box.

Not all ML models can directly use previous values of the target variable to predict the next steps because of that, so-called lag features were considered as well. Lag n of a feature or a target variable is essentially a corresponding value n steps back into history, e.g., lag 7 of electricity generation for today will be the value of generation 7 days ago. The second part of this section contains details on the cases in which these features were or were not utilized.

Given the prepared dataset of historical data, one can train an ML model using it. This results in an estimator that effectively maps input features to the target variable with a certain degree of quality. This quality can be measured using certain evaluation metrics calculated by comparing the actual target values with predicted ones. Considering the nature of the problem considered in solar energy generation forecasting, it was decided to evaluate and compare models by Root Mean Squared Error (RMSE) metric values, as it heavily penalizes significant deviations of the forecast from actual values. This particularity is essential for reflecting the quality of the forecast. In addition, it has the same dimensionality as the forecasted target variable, which makes results easier to interpret. However, to allow for model comparison, one has to make sure that the models were trained in the same setup, e.g., the same training data was available to the model training process, and the evaluation metrics were calculated on the same subsets of the data. This allows for a fair comparison and enables making conclusions based on the changes in the evaluation metrics value. For instance, certain model types are more applicable to specific forecasting problems.

Moreover, given that the experiment setup is the same for several experiments, one can make changes in one of the experiment parameters and make certain conclusions based on the dependency of the model performance metric and the experiment parameter values.

It is important to emphasize that to draw conclusions based on such dependency, it is crucial to have at least somewhat robust estimates of the model's performance. To this end, several best practices can be utilized. First, one has to make sure that the model performance evaluation is done using the data that was not available to the model during the training process to avoid so-called data leakage. A particular case of this happens when the data has a time dimension in it, as data leakage from the future to the past can happen, although the feature set itself is valid. Hence, it is advised to use a classic train-test data split to avoid such issues. It is a technique that splits data into train and test subsets of the data. The training part is used for model training, while the test is used for performance evaluation only. A time series cross-validation technique can also be applied for a more robust and reliable model performance estimation. In a cross-validation process, the training data is subsequently split into several combinations of the training and validation subsets - folds. This approach allows the model training and validation process to be repeated several times. The final model performance estimate is then taken as some aggregation, usually average, of model performance on validation data folds. Time series stands for a particular design where target variable information is not leaking from fold to fold by design (See Fig. 3). This is often combined with hyperparameter tuning.

In such cases, a hold-out test data part is still required, as all data samples in the cross-validation process are used either explicitly in the training process or implicitly to choose optimal hyperparameters of the model.

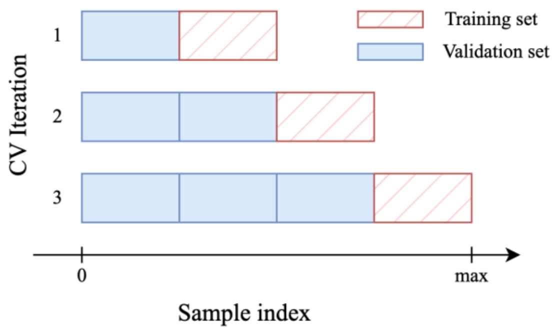


Figure 3. Time series cross-validation technique

As manual tuning is not considered a reliable way to get the best hyperparameter combinations for a specific model type, cross-validation can be employed with a hyperparameter tuning approach. Random Search is regarded as a better alternative to the straightforward Grid Search [44].

A wide variety of the existing ML models were considered, as this paper focuses on studying the particularities of their application to the considered task of decision support and underlying solar power forecasting, not the attempt to create the best-performing model for specific forecasting datasets. Because of that, several classes of the models were considered:

1. linear models, including Lasso and Ridge modifications [45];
2. simple tree-based models, such as Random Forest [45];
3. gradient boosting-based tree ensembles, such as GradientBoostingRegressor [45] or LightGBM [46];
4. Support Vector Regressor (SVR) [45] - an adaptation of the SVM for regression tasks;
5. multi-layered perceptron (MLP) [45] - a model type with a much bigger capacity, in theory;
6. time-series approaches, such as SARIMA [47].

Therefore, to answer the research question outlined in Problem Statement Section, different ML models should be trained using train-test splits, time series cross-validation, and hyperparameters tuning techniques. Their corresponding performance and the dependency of that performance on the experiment parameters reflecting the research questions will allow for finding answers to the questions.

It is important to clarify the definition of prediction horizon to avoid ambiguity in answering the second research question. In this work, the prediction horizon is the number of steps ahead in the future that a model should predict without updating its knowledge of the historical data. In some use cases, like the one considered in this work, previous target variable values are used as lag features and, consequently, as an input to the model prediction process.

Depending on the prediction horizon, it may or may not be valid to use actual historical values in the evaluation process. For instance, when using lag one and prediction horizon of one, the model makes one prediction step at a time, and then historical data can be updated. This way, it’s always valid to use actual historical values for the lag one feature.

However, this will not be the case if the prediction horizon is higher than the lag, e.g., lag one and prediction horizon of seven. The model should make seven prediction steps before updating the historical data, and it can’t use lag one feature filled with actual historical values from the previous day. Then, the model's prediction from the previous step within the horizon should be used as the corresponding lag feature. For example, a model makes the first prediction of 2.5.

Then, for the next prediction step, the model should use this value as a lag one feature value instead of an actual historical lag value.

The implementation and execution of the experiment pipelines were done on an Intel 7-based machine with a Windows 10 OS in the Python 3.10 environment with the following major packages: scikit-learn (1.2.2) [45], scipy (1.10.1) [48], pmdarima (2.0.4) [47] , pandas (2.2.2) [49], numpy (1.24.3) [50], matplotlib (3.9.2) [51], lightgbm (4.5.0) [46].

A set of experiments must be done to

identify the model type that best fits the considered problem. To this end, a high-level comparison of the models can be done to determine the model class to be used for more detailed experiments, e.g., with more fine-tuning and changing experiment parameters that reflect the research questions from the Problem Statement Section.

In this work, 5% of the data, which is chronologically latest, was used for test purposes if otherwise not stated for a particular experiment.

First, classic time series approaches had to be considered. Manual tuning of the SARIMA model parameters is not entirely reliable, and an automated tuning process was also utilized to find the best SARIMA model for the given data.

Second, other model types were considered. For the sake of simplicity, regression models available in the sci-kit-learn library were considered. This way, the experiment pipeline can be unified and reused for all the models. The list of considered models is as follows: LinearRegression, Ridge, Lasso, SVR, MLPRegressor, RandomForestRegressor, GradientBoosting-Regressor, and ExtraTreesRegressor [45]. These models provide a good representation of different ML model types.

For each model type, several hyperparameter combinations were considered. The idea was that some hyperparameter variation would improve our understanding of the model performance for the given data. Hyperparameter combinations that we considered mainly were one of three following types:

– Commonly used hyperparameter combination that should work reasonably well in most cases for the given model type. One can think of it as default hyperparameters.

– Hyperparameter combination that allows a model to overfit to the training data. This allows us to evaluate model capacity in favorable conditions.

– Hyperparameter combination with strong regularization components that should stop overfitting and improve test metrics even if the train metrics value would suffer from this.

This way, the model type's behavior can be understood better in relation to the given problem and dataset.

It is important to mention an essential difference between SARIMA and other considered model types. SARIMA explicitly uses previous target variable values to predict the next steps but doesn't have a direct ability to use other available features.

For other models, it's precisely the opposite. Because of that, we first compared the performance of the time series approach with the Scikit-learn models without lags but then added lags and compared the performance with the most prominent model type.

Lag features derived from the target variable were used, with lags of 1,3,5,7.

After comparing model behavior and performance, gradient-boosting tree-based models were selected to continue with the main experiments of this research work. As the following experiments required extensive tuning of the hyperparameters, it was decided to stick to the LightGBM implementation, known for its blazing-fast performance and light model footprint [46].

Next, the following setups were suggested to answer the research questions outlined in Problem Statement:

1. Set parameters for the experiment, such as ranges of hyperparameters, input dataset, number of folds in the time series cross-validation, hold-out test data size, number of tuning iterations, etc.
2. Conduct hyperparameter tuning using the Random Search approach, given hyperparameter ranges, and train part of the data. Time series cross-validation with three folds was used to get a robust estimate of model performance for each hyperparameter combination.
3. Evaluate the best model on the test data part.
4. Change one parameter of the experiment and repeat the procedure.
5. Gather model performances for each parameter value and make conclusions about the dependency of the parameter on the model application. The final best model received as a result of full tuning in each experiment was used to compare results.

As all experiment parameters are set,

except for the one that is being studied, one can attribute changes in model performance to the change in that one experiment parameter.

After having the experiment procedure implemented, several parameters of the experiment were varied to understand the implications for the considered research questions:

– size of the available training data;
– size of the testing data;
– prediction horizon.

The best model was received with 1000 iterations of the hyperparameters tuning process.

Along with the experiment logs, prediction charts and model files were saved. These allow for making a qualitative evaluation of predictions compared to the actual historical data, as well as comparing model file sizes.

The latter can be utilized to answer the last research question from the Problem Statement on model size considerations.

Results

Table 1 summarizes performance evaluation metrics, namely RMSE, for both train and test data subsets. Lower values of RMSE indicate higher performance.

Table 1. Best performance per model type

Model type	TrainRMSE	TestRMSE
Linear Regression	1.6	2.52
Ridge	1.6	2.52
Lasso	1.82	2.34
SVR	1.86	2.62
MLPRegressor	1.39	3.47
RandomForest Regressor	1.32	2.59
Gradient Boosting Regressor	1.21	2.67
ExtraTrees Regressor	0.76	2.77
RandomForest Regressor with lags	1.27	2.31
SARIMA	-	2.31

It is observed that performance varies significantly among model classes. Some of the more complex models, like SVR and MLP, do not demonstrate performance superiority. The best test RMSE values are demonstrated by RandomForest and SARIMA models. Linear models show good test RMSE values but comparatively worse performance in terms of train metric values.

Table 2 contains performance metric values for the best LightGBM model received after tuning for different amounts of the training data that were available to the model tuning process. It can be seen that with more training data, performance on the test data improves while train RMSE increases.

Table 2. Tuned LightGBM model performance for different training data sizes

Training data size	Train RMSE	Test RMSE
Full data (2020-2023)	1.4	2.03
2021-2023	1.21	2.16
2022-2023	1.36	2.34
2023 only	0.18	2.56

LightGBM model performance dependence on the test data size is presented in Table 3.

The test size was changed in this experiment, but the same prediction horizon of one step (one day) ahead was utilized. In this case, the lag features used as input in each prediction step are calculated on the actual historical data. This corresponds to the situation when forecasting is done for the next day. Hence, the actual values of a target variable from previous days are used. A DSS can collect this data during its work. In other words, lag features don't contain any model predictions. It was observed that model performance fluctuates with the change in test sample size without a clear monotonous pattern.

Table 3. Tuned LightGBM model performance for different test data sizes

Test size	Train RMSE	Test RMSE
5% of the full data - 71	1.4	2.03
30	1.42	2.15
14	1.42	2.05
7	1.43	2.20

Table 4 presents the performance

dependence on the prediction horizon size. It was observed that the training metric remains stable, as expected. On the other side, test RMSE increases with the increase of horizon size.

Table 4. Tuned LightGBM model performance for different prediction horizons

Horizon size	Train RMSE	Test RMSE
5% of the full data - 71	1.4	2.55
30	1.4	2.53
14	1.4	2.35
7	1.4	2.24
1	1.4	2.03

Observed model sizes per model type are listed in Table 5. It is observed that the size of simpler linear models is stable and small. Especially if one compares to tree-based models that are not only much bigger but also grow in size depending on the model structure (e.g., the number of trees in a forest).

Table 5. Model file size in megabytes for different model types

Model type	Pickle-file size, MB
LinearRegression	[0.001]
Ridge	[0.001, 0.001, 0.001]
Lasso	[0.001, 0.001, 0.001]
SVR	[0.116, 0.125, 0.11]
MLPRegressor	[0.025, 0.036, 0.151]
RandomForest Regressor	[0.199, 3.897, 18.968]
GradientBoosting Regressor	[0.115, 0.682, 44.274]
ExtraTreesRegressor	[4.804, 30.542, 45.182]
LightGBM Regressor	[0.001, 0.001, 0.001]

However, as shown in Table 6, which shows model performance to model file size dependency, there is no clear relation between bigger and more complex models providing a higher level of performance.

Table 6. Model file size in megabytes and test RMSE for RandomForest models

Model performance (Test RMSE)	Model file size, MB
2.6	0.199
2.73	3.897
2.77	18.968

Discussion

The first set of experiments was focused on choosing the model type for further experiments. It became evident that some model types are more prone to overfitting, while others lack the capacity to learn complex patterns. In particular, it is important to emphasize that linear models provide a comparable level of performance to tree-based models. However, they demonstrated limited capacity to improve the evaluation metric further. This limitation is due to the linearity assumption at the core of such model design. Thus, tree-based models were chosen as a more flexible option for the following experiments. However, to account for model size concerns, it was decided to consider a more efficient implementation from the LightGBM library instead of standard scikit-learn implementations of various tree-based models. It is also worth mentioning that a neural network model (MLP) may be capable of demonstrating a better level of performance, but this would likely require significant efforts for tuning not only hyperparameters but also the model structure (i.e. the number of layers and neurons in the network architecture).

It was observed (see Table 2) that there is a direct relation between the amount of training data and model performance on unseen data. Train performance improves with a smaller amount of data, most likely, as in such cases, it is easier for the model to fit the training data. However, the test performance drops significantly after decreasing the amount of training data to below two full years. Thus, in practical applications, it is advised to make sure training data contains at least two full seasonal periods. For instance, in the

considered applied problem, there is strong evidence of yearly seasonality, and thus, two full years of training data are required to let the model learn the yearly seasonal patterns.

Table 3 demonstrates that model performance fluctuates if the test period size is changed. It can be concluded that the length of the test period should be chosen based on the applied problem context and reflect the underlying processes' particularities. It is worth mentioning that the test period length should be selected before the experiment to allow for correct experiment procedure and reliable results.

It was observed that there is a clear dependency of the performance on the prediction horizon (see Table 4). The bigger the horizon, the higher the RMSE value for the test period. This dependency reflects a decrease in model performance when its own predictions are utilized as input for predicting the sequential steps. Hence, the uncertainty of inputs increases, and the model has less reliable information to make precise predictions. This result indicates that in practical applications, it is better to choose the shortest applicable horizon of forecasting to reduce the uncertainty of model inputs and allow for better performance by maximizing the probability of a precise forecast. In the case of DSSs deployed on edge to prosumer's compute and operating on a daily granularity level, it should be feasible to make one-day ahead predictions. This doesn't mean a bigger number of prediction steps should always be avoided in practical applications. Rather, that there is a clear benefit in updating the forecast using the newly available model input values as frequently as possible from a practical perspective.

As shown in Table 5, different model types have different sizes. However, there is no explicitly observed dependency between the model's size and its performance. For example, linear models are the smallest but have limited capacity, while LightGBM regression models have approximately the same size and better performance. At the same time, other model types demonstrated lower performance and orders of magnitude bigger file sizes. Moreover, as was investigated for RandomForest models, more complex models are not directly linked to a better performance. For instance, a model with a high number of deep trees will have a bigger footprint but is likely to overfit and show modest performance on the hold-out test dataset. This should be considered as a counterexample, not as a general rule, though.

Considering model size and performance, such a model is feasible for local deployment even on a modest prosumer computing infrastructure, as limited resources are required to make predictions. There is no requirement for real-time performance, and daily forecast updates are totally feasible.

Conclusions

In this work, several aspects of applying ML approaches to forecasting in the context of prosumer decision support were investigated to provide practical recommendations for the development of such applied solutions, such as the required amount of training data and prediction horizon size. Based on the considered use case, it was observed that it is beneficial to apply models with significant capacity while controlling their tendency to overfit. In the case of time-related data and a model type that doesn't take previous values into account explicitly, it is also worth considering adding corresponding features reflecting previous variable values. For the training data, one should aim to gather data for at least two full seasonality periods to allow the model to learn the seasonality pattern. The prediction horizon should be as short as the applied problem allows, especially in the cases when predictions of the model have to be included in the inputs of the following step prediction process inside the prediction horizon, as this helps to limit uncertainty multiplication in the forecasting process.

The scientific novelty of the obtained results is that the recommendations regarding the ML application to the forecasting problem were formulated based on the real-world scenario with daily forecasting frequency. Moreover, it was considered in the context of decision-making support for a prosumer at the level of individual household size. The corresponding applied experiments justify these recommendations and allow for better

development of the forecasting solutions in the energy prosumer-related use cases.

The practical significance of the results consists of two components. First, the results were obtained from an applied forecasting problem using real-world data, where the entire experiment pipeline was implemented and the experiments conducted. Second, results are general applicability due to experiments conducted using real-world data from a specific prosumer use case. This way, the derived suggestions on the application of ML models to the daily forecasting problem are relevant to similar use cases and can be employed there to improve. In particular, the considered use case includes additional uncertainty from other parts of the system, which makes it different from the purely theoretical studies, e.g. considered historical data suffers from missing values and gap periods the same way as may happen in various real-world scenarios.

Prospects for future research are to extend the number of considered decision-making support scenarios and use cases. This extension may allow for the extraction of general patterns in the ML models application and give both general rules for different scenarios and specific recommendations for them, e.g. daily and hourly forecasting.

One specific research direction that may result in a universal tool applicable to various use cases would be automated detection of the recommended minimum training period and/or prediction horizon based on the available historical data, specific problem parameters, and business problem constraints.

Lastly, it may be possible to achieve different results using more advanced models and significant efforts to improve their performance in each specific use case, e.g. combining attention-based, tree-based, and time series trend models in a sophisticated system that may shift recommendations for the amount of training data, evaluation methodology or prediction horizon. Thus, any work that aims at a generic solution for a certain type of use case would be of significant value to the research and industrial community.

Acknowledgements

The work is supported by the state budget scientific research project of Sumy State University Intelligent information technology for proactive management of energy infrastructure in conditions of risk and uncertainty” (state registration number 0123U101852).

References

1. Alotaibi, I., Abido, M. A., Khalid, M., & Savkin, A. V. (2020). A Comprehensive Review of Recent Advances in Smart Grids: A Sustainable Future with Renewable Energy Resources. Energies, 13(23), 6269. https://doi.org/10.3390/en13236269
2. Benti, N. E., Chaka, M. D., & Semie, A. G. (2023). Forecasting Renewable Energy Generation with Machine Learning and Deep Learning: Current Advances and Future Prospects. Sustainability, 15(9), 7087. https://doi.org/10.3390/su15097087
3. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13.
4. Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. WIREs Data Mining and Knowledge Discovery, 13(2). https://doi.org/10.1002/widm.1484
5. Chomać-Pierzecka, E., Rogozińska-Mitrut, J., Różycka, M., Soboń, D., & Stasiak, J. (2023). Energy Innovation for Individual Consumers in Poland – Analysis of Potential and Evaluation of Practical Applications in Selected Areas. Energies, 16(15), 5766. https://doi.org/10.3390/en16155766
6. Costa, R. L. de C. (2022). Convolutional-LSTM networks and generalization in forecasting of household photovoltaic generation. Engineering Applications of Artificial Intelligence, 116, 105458. https://doi.org/10.1016/j.engappai.2022.105458
7. Dimeas, A., Drenkard, S., Hatziargyriou, N., Karnouskos, S., Kok, K., Ringelstein, J., & Weidlich,A. (2014). Smart Houses in the Smart Grid: Developing an interactive network. IEEE Electrification Magazine, 2(1), 81–93. https://doi.org/10.1109/MELE.2013.2297032
8. Dorado-Rojas, S. A., Bogodorova, T., & Vanfretti, L. (2021). Time Series-Based Small-Signal Stability Assessment using Deep Learning. 2021 North American Power Symposium (NAPS), 1–6. https://doi.org/10.1109/NAPS52732.2021.9654643
9. Gan, L., Jiang, P., Lev, B., & Zhou, X. (2020). Balancing of supply and demand of renewable energy power system: A review and bibliometric analysis. Sustainable Futures, 2, 100013. https://doi.org/10.1016/j.sftr.2020.100013
10. Grimaldo, A. I., & Novak, J. (2020). Combining Machine Learning with Visual Analytics for Explainable Forecasting of Energy Demand in

Prosumer Scenarios. Procedia Computer Science, 175, 525–532. https://doi.org/10.1016/j.procs.2020.07.07

11. Gržanić, M., Capuder, T., Zhang, N., & Huang, W. (2022). Prosumers as active market participants: A systematic review of evolution of opportunities, models and challenges. Renewable and Sustainable Energy Reviews, 154, 111859. https://doi.org/10.1016/j.rser.2021.111859

12. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

13. Hu, J.-L., & Chuang, M.-Y. (2023). The Importance of Energy Prosumers for Affordable and Clean Energy Development: A Review of the Literature from the Viewpoints of Management and Policy. Energies, 16(17), 6270. https://doi.org/10.3390/en16176270

14. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

15. Jing, X., Song, M., Gao, C., Wang, C., Li, L., & Liu, W. (2022). Analysis of the decision-making process of prosumers in the transactive energy market : From the perspective of traditional economics and behavioral economics. 2022 IEEE 5th International Electrical and Energy Conference (CIEEC), 4679–4684. https://doi.org/10.1109/CIEEC54735.2022.9846116

16. Jozi, A., Pinto, T., Praça, I., & Vale, Z. (2019). Decision Support Application for Energy Consumption Forecasting. Applied Sciences, 9(4), 699. https://doi.org/10.3390/app9040699

17. Karande, P., Gallagher, B., & Han, T. Y.-J. (2022). A Strategic Approach to Machine Learning for Material Science: How to Tackle Real-World Challenges and Avoid Pitfalls. Chemistry of Materials, 34(17), 7650–7665. https://doi.org/10.1021/acs.chemmater.2c01333

18. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 2017-December.

19. Koltsaklis, N., Panapakidis, I. P., Pozo, D., & Christoforidis, G. C. (2021). A Prosumer Model Based on Smart Home Energy Management and Forecasting Techniques. Energies, 14(6), 1724. https://doi.org/10.3390/en14061724

20. Lopes, J. A. P., Madureira, A. G., Matos, M., Bessa, R. J., Monteiro, V., Afonso, J. L., Santos, S. F., Catalão, J. P. S., Antunes, C. H., & Magalhães, P. (2020). The future of power systems: Challenges, trends, and upcoming paradigms. WIREs Energy and Environment, 9(3). https://doi.org/10.1002/wene.368

21. Lukianykhin, O., & Bogodorova, T. (2021). Voltage Control-Based Ancillary Service Using Deep Reinforcement Learning. Energies, 14(8), 2274. https://doi.org/10.3390/en14082274

22. Lukianykhin, O., Shendryk, V., Shendryk, S., & Malekian, R. (2024). Promising AI Applications in Power Systems: Explainable AI (XAI), Transformers, LLMs. In International Conference "New Technologies, Development and Applications" (pp. 66–76). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-66271-3_8

23. McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 56–61. https://doi.org/10.25080/Majora-92bf1922-00a

24. Mellit, A., Massi Pavan, A., Ogliari, E., Leva, S., & Lughi, V. (2020). Advanced Methods for Photovoltaic Output Power Forecasting: A Review. Applied Sciences, 10(2), 487. https://doi.org/10.3390/app10020487

25. Melo, F. C., Carrilho da Graça, G., & Oliveira Panão, M. J. N. (2023). A review of annual, monthly, and hourly electricity use in buildings. Energy and Buildings, 293, 113201. https://doi.org/10.1016/j.enbuild.2023.113201

26. Mohammadi, E., Alizadeh, M., Asgarimoghaddam, M., Wang, X., & Simoes, M. G. (2022). A Review on Application of Artificial Intelligence Techniques in Microgrids. IEEE Journal of Emerging and Selected Topics in Industrial Electronics, 3(4), 878–890. https://doi.org/10.1109/JESTIE.2022.3198504

27. Muttaqi, K. M., Islam, Md. R., & Sutanto, D. (2019). Future Power Distribution Grids: Integration of Renewable Energy, Energy Storage, Electric Vehicles, Superconductor, and Magnetic Bus. IEEE Transactions on Applied Superconductivity, 29(2), 1–5. https://doi.org/10.1109/TASC.2019.2895528

28. Natarajan, V. A., & Karatampati, P. (2019). Survey on renewable energy forecasting using different techniques. 2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC), 349–354. https://doi.org/10.1109/ICPEDC47771.2019.9036569

29. Olson, R. S., Cava, W. La, Mustahsan, Z., Varik, A., & Moore, J. H. (2018). Data-driven advice for applying machine learning to bioinformatics problems. Biocomputing 2018, 0(212669), 192–203. https://doi.org/10.1142/9789813235533_0018

30. Panapakidis, I. P., Koltsaklis, N. E., & Christoforidis, G. C. (2021). Forecasting Methods to Support the Decision Framework of Prosumers in Deregulated Markets. 2021 9th International Conference on Modern Power Systems (MPS), 1–5. https://doi.org/10.1109/MPS52805.2021.9492725

31. Parag, Y., & Sovacool, B. K. (2016). Electricity market design for the prosumer era. Nature Energy, 1(4), 16032. https://doi.org/10.1038/nenergy.2016.32

32. Parra-Domínguez, J., Sánchez, E., & Ordóñez, Á. (2023). The Prosumer: A Systematic Review of the New Paradigm in Energy and Sustainable Development. Sustainability, 15(13), 10552. https://doi.org/10.3390/su151310552

33. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Cournapeau,

D., Brucher, M., & Perrot, M. (2011). Scikit-learn: Machine Learning in Python Pedregosa, Varoquaux, Gramfort et al. Journal of Machine Learning Research, 12. https://doi.org/10.5555/1953048.2078195

34. Peng, F. Z., Liu, C.-C., Li, Y., Jain, A. K., & Vinnikov, D. (2024). Envisioning the Future Renewable and Resilient Energy Grids – A Power Grid Revolution Enabled by Renewables, Energy Storage, and Energy Electronics. IEEE Journal of Emerging and Selected Topics in Industrial Electronics, 5(1), 8–26. https://doi.org/10.1109/JESTIE.2023.3343291

35. Petrican, T., Vesa, A. V., Antal, M., Pop, C., Cioara, T., Anghel, I., & Salomie, I. (2018). Evaluating Forecasting Techniques for Integrating Household Energy Prosumers into Smart Grids. 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP), 79–85. https://doi.org/10.1109/ICCP.2018.8516617

36. Piekut, M. (2021). The Consumption of Renewable Energy Sources (RES) by the European Union Households between 2004 and 2019. Energies, 14(17), 5560. https://doi.org/10.3390/en14175560

37. Pirbazari, A. M. (2021). Predictive Analytics for Maintaining Power System Stability in Smart Energy Communities. In Predictive Analytics for Maintaining Power System Stability in Smart Energy Communities. University of Stavanger. https://doi.org/10.31265/usps.83

38. Priestley, M., O’donnell, F., & Simperl, E. (2023). A Survey of Data Quality Requirements That Matter in ML Development Pipelines. Journal of Data and Information Quality, 15(2), 1–39. https://doi.org/10.1145/3592616

39. Rahman, M. S., Khomh, F., Hamidi, A., Cheng, J., Antoniol, G., & Washizaki, H. (2023). Machine learning application development: practitioners’ insights. Software Quality Journal, 31(4), 1065–1119. https://doi.org/10.1007/s11219-023-09621-9

40. Rai, A., Shrivastava, A., & Jana, K. C. (2023). An Empirical Analysis of Machine Learning Algorithms for Solar Power Forecasting in a High Dimensional Uncertain Environment. IETE Technical Review, 40(4), 558–573. https://doi.org/10.1080/02564602.2022.2136270

41. Scarabaggio, P., Grammatico, S., Carli, R., & Dotoli, M. (2022). Distributed Demand Side Management With Stochastic Wind Power Forecasting. IEEE Transactions on Control Systems Technology, 30(1), 97–112. https://doi.org/10.1109/TCST.2021.3056751

42. Schultz, B. G., Joukhadar, Z., Nattala, U., Quiroga, M. del M., Bolk, F., & Vogel, A. P. (2021). Best practices for supervised machine learning when examining biomarkers in clinical populations. In Big Data in Psychiatry #x0026; Neurology (pp. 1–34). Elsevier. https://doi.org/10.1016/B978-0-12-822884-5.00013-1

43. Smith, T. G. (2017). pmdarima: ARIMA estimators for Python. https://www.alkaline-ml.com/pmdarima/

44. Sobri, S., Koohi-Kamali, S., & Rahim, N. Abd. (2018). Solar photovoltaic generation forecasting methods: A review. Energy Conversion and Management, 156, 459–497. https://doi.org/10.1016/j.enconman.2017.11.019

45. Solea, C., Hera, C., Florea, A., Morariu, D., Stanescu, D., & Vintan, M. (2024). A Comparative Study on Prosumers Load Demand and Production Forecasting Using Machine Learning and Time Series Techniques. 2024 International Conference on Applied and Theoretical Electricity (ICATE), 1–7. https://doi.org/10.1109/ICATE62934.2024.10749434

46. Soman, S. S., Zareipour, H., Malik, O., & Mandal, P. (2010). A review of wind power and wind speed forecasting methods with different time horizons. North American Power Symposium 2010, 1–8. https://doi.org/10.1109/NAPS.2010.5619586

47. Taik, A., Nour, B., & Cherkaoui, S. (2021). Empowering Prosumer Communities in Smart Grid with Wireless Communications and Federated Edge Learning. IEEE Wireless Communications, 28(6), 26–33. https://doi.org/10.1109/MWC.017.2100187

48. Teixeira, R., Cerveira, A., Pires, E. J. S., & Baptista, J. (2024). Advancing Renewable Energy Forecasting: A Comprehensive Review of Renewable Energy Forecasting Methods. Energies, 17(14), 3480. https://doi.org/10.3390/en17143480

49. Triantafyllou, A., Jimenez, J. A. P., Torres, A. D. R., Lagkas, T., Rantos, K., & Sarigiannidis, P. (2020). The Challenges of Privacy and Access Control as Key Perspectives for the Future Electric Smart Grid. IEEE Open Journal of the Communications Society, 1, 1934–1960. https://doi.org/10.1109/OJCOMS.2020.3037517

50. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Vázquez-Baeza, Y. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods, 17(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

51. Xie, Y., Li, C., Li, M., Liu, F., & Taukenova, M. (2023). An overview of deterministic and probabilistic forecasting methods of wind energy. IScience, 26(1), 105804. https://doi.org/10.1016/j.isci.2022.105804

The article has been sent to the editors 12.03.25.
After processing 22.03.25.
Submitted for printing 30.03.25.